AD-A279 219

ARL-TN-49

AR-008-375

(2)

DEPARTMENT OF DEFENCE

DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION

AERONAUTICAL RESEARCH LABORATORY

MELBOURNE, VICTORIA

Technical Note 49

PROGRAMMABLE COCKPIT RESEARCH SIMULATOR

by

M. IOB
A.G. PAGE
D.A. CRAVEN

Approved for public release

MARCH 1994

AR-008-375

DEPARTMENT OF DEFENCE
DEFENCE SCIENCE AND TECHNOLOGY ORGANISATION
AERONAUTICAL RESEARCH LABORATORY

Technical Note 49

PROGRAMMABLE COCKPIT RESEARCH SIMULATOR

by

M. IOB
A.G. PAGE
D.A. CRAVEN

SUMMARY

The 'Programmable Cockpit' is a low-cost facility to be used for study and development of the aircrew-vehicle interface for future aircraft systems. It was designed so that instrument layouts and display formats could be reconfigured rapidly and tested in a realistic aircraft representation or emulation, with the pilot under representative workload conditions. It is not intended to be used for pilot training and/or evaluation.

It uses personal computers and computer graphics workstations linked together to represent the aircraft displays, and includes representative flight dynamic models for fixed-wing aircraft. Controls include a sidestick, rudder pedals, throttle, and touch sensitive screens.

This document gives a brief overview of the complete Programmable Cockpit, including the controller station (which incorporates the flight dynamic model), inter-computer communications, rapid prototyping environment and display rehosting.

DSTO
AUSTRALIA

POSTAL ADDRESS: **Director, Aeronautical Research Laboratory**
506 Lorimer Street, Fishermens Bend
Victoria 3207, Australia.

CONTENTS	Page No.

EXECUTIVE SUMMARY

Multifunction electronic displays in modern aircraft provide great flexibility in information presentation; crew stations can be reconfigured by software changes and tailored to specific requirements. The processing power of flight-management/mission computers is increasing constantly and is generally not a limiting factor. There are opportunities for totally new concepts in display content, which in the military situation may offer significant increases in operational effectiveness. Failure to integrate a new capability properly, with due consideration for the total cockpit 'ecology', can limit the effectiveness of the system particularly in the case of a single pilot operation. In Air Operations Division the need for a low cost facility for research into novel display concepts, and their effectiveness in information transfer, was identified. Objective investigation of effectiveness of new displays required a human-in-the-loop facility.

This paper briefly describes the results of a project to provide a low-cost cockpit simulation facility intended for evaluation of flight information integration and human-machine interface design. An important feature of the facility is that it can be easily and quickly reconfigured to represent a new design and is capable of supporting a wide range of different designs, hence the name 'Programmable Cockpit' (PC). This facility is now operational and forms one part of the Air Operations Simulation Centre (AOSC) in Air Operations Division.

The PC is essentially built around a set of video screens driven by interconnected personal computers and graphics workstations (i.e. via Ethernet). The computers read control inputs, iterate the flight dynamic equations and generate the graphics to display the instruments and controls. Some of the video screens are touch-sensitive so that switches, buttons etc. drawn on the screens can be made active. Rapid prototyping software is used to generate displays on the workstations and for the personal computers. All other software necessary for its operation is written in high-level languages so that changes can be implemented quickly and reliably.

The PC has already been used for a number of applications. Cockpit instruments from the PC9/A and F-111C aircraft have been emulated using the rapid prototyping capability and these can be used in isolation or as part of a complete cockpit representation. Some perspective highway-in-the-sky and Flight Director display formats, for research into effectiveness of displays for assisting in trajectory control (as in precision curved landing approaches), were developed. Other cockpit instruments were selected to enable the Flight Director to be incorporated into a cockpit for preliminary evaluation of novel features of this type of display. The PC has also been used to design and develop some new Radar Warning Receiver display formats for the F-111C. We intend to use the PC to evaluate these display concepts in a part-task simulation.

The Programmable Cockpit has many applications, from the development of novel display techniques, to prototyping and evaluating modifications to existing display instruments. It can also be used as an "idea filter" for larger, more expensive simulators. Display ideas can be prototyped in the PC environment, in a relatively short time and at a low cost. Ideas that show some potential and require further investigation could then be ported onto a higher fidelity research simulator within the AOSC. Alternatively it could act as a subsiduary pilot input station in a multi-pilot simulation in the AOSC.

1.0 INTRODUCTION

Multifunction electronic displays in modern aircraft provide great flexibility in information presentation; crew stations can be reconfigured by software changes and tailored to specific requirements. The processing power of flight-management/mission computers is increasing constantly and is generally not a limiting factor. There are opportunities for totally new concepts in display content, which in the military situation may offer significant increases in operational effectiveness. Failure to integrate a new capability properly, with due consideration for the total cockpit 'ecology', can limit the effectiveness of the system particularly in the case of a single pilot operation. In Air Operations Division the need for a low cost facility for research into novel display concepts, and their effectiveness in information transfer, was identified. Objective investigation of effectiveness of new displays required a human-in-the-loop facility.

This paper briefly describes the results of a project to provide a low-cost cockpit simulation facility intended for evaluation of flight information integration and human-machine interface design. An important feature of the facility is that it can be easily and quickly reconfigured to represent a new design and is capable of supporting a wide range of different designs, hence the name 'Programmable Cockpit' (PC).

The first, earlier, stage of the Programmable Cockpit (PC-1) was constructed from easily available general-purpose components (see references 1, 2, 3 and 4). It was intended mainly to demonstrate the potential of the concept. However, it was found that the update rate of the primary flight instruments, such as the Attitude Indicator, was too slow to permit proper investigation of flight control issues. In addition, the acquisition of a rapid prototyping capability was considered essential if the development time for new designs were to be reduced to acceptable levels.

An upgraded second stage development (PC-2) is now operational incorporating faster computers and rapid prototyping software. It has been incorporated into the Air Operations Simulation Centre (AOSC) of Air Operations Division. The PC-2 is making it possible to perform human-in-the-loop research.

2.0 PROGRAMMABLE COCKPIT OVERVIEW

The PC-2 is essentially built around a set of video screens driven by interconnected personal computers and graphics workstations. The computers read control inputs, iterate the flight dynamic equations and generate the graphics to display the instruments and controls. Some of the video screens are touch-sensitive so that switches, buttons etc. drawn on the screens can be made active. Rapid prototyping software is used to generate displays on the workstations and for the personal computers. All other software necessary for its operation is written in high-level languages so that changes can be implemented quickly and reliably.

A maximum of four Commodore Amiga[1] and two Silicon Graphics[2] IRIS generated displays are used in the cockpit although more could conceivably be used. The Silicon Graphics machines were acquired for their high speed and high resolution graphics capability and they are used for the more dynamic cockpit displays. The Commodore Amiga computers possess a good graphics capability which is adequate for less dynamic cockpit displays. The number and positioning of the screens can be selected to reflect the requirements of the particular simulation. They have been re-boxed to minimise the space around the active screen area, and the new housings are rectangular so they can be stacked or rearranged into whatever layout is appropriate. One of the workstations can be configured to provide a Head Up display and/or an outside world display although a

1. Commodore and Amiga are trademarks of Commodore Business Machines Inc., West Chester, Pennsylvania, USA.
2. Silicon Graphics Inc., Mountain View, California, USA.

projector could be used for outside views rather than a monitor. Another Amiga is used as the controller station. All these machines are connected via Ethernet for data communication. Other forms of communication such as serial or parallel can be used if necessary for a particular configuration. The layout is shown in Figure 1 and the hardware is detailed in the appendix.

2.1 Controller Station

The controller station is essentially a mouse-driven graphical user interface and an aircraft flight dynamic model (AFDM) which is controlled from the interface. A full six degree-of-freedom aircraft model is used employing simple engine models. The aircraft control inputs (throttle, sidestick and rudder pedal) are fed into the model via an analog-to-digital converter. The AFDM determines the aircraft's response and updates the Ethernet communications link with all the parameters used by other sections of the PC-2. The graphical interface gives the operator control over flight conditions and cockpit configuration. Refer to section 3.0 for more detail.

2.2 Rapid Prototyping Software

A suite of rapid prototyping software has been purchased which is hosted on one of the IRIS workstations. This software provides the capability to design functional graphics displays and controls using an interactive mouse-driven interface. New designs can be generated in a relatively short time for evaluation in the PC-2 environment. Modifications to a prototype can be made quickly and easily enabling rapid iteration through the design process. More detail on the rapid prototyping environment can be found in section 4.0.

2.3 Ethernet Communications

The Ethernet communications software employs the Transmission Control Protocol/Internet Protocol (TCP/IP) for data transmission between the various elements of the PC-2. At a higher level the software uses the protocols employed by the rapid prototyping software for data communications. These protocols have been adopted as the standard for all inter-computer communications in the system although the presence of the rapid prototyping software is not necessary for the rest of the simulation to function. The software consists of modules which are incorporated into the programs that are executing on each of the machines in any given PC-2 configuration. Section 5.0 discusses this software in greater detail.

2.4 Display Rehosting

A software tool has been created which enables displays generated with the rapid prototyping software to be rehosted to an Amiga computer. The rapid prototyping software produces a 'Graphics Metafile' which is used by the rehosting tool to produce a graphics display program for an Amiga. Using this tool many different displays can be easily rehosted to Amiga computers. The rehosting software is discussed in greater detail in section 6.0.

3.0 CONTROLLER STATION

The controller station is hosted on an Amiga computer separate from the cockpit allowing an operator to control the PC-2 simulation. It incorporates the flight dynamic model described in reference 1 although the model has been ported from the IBM PC-AT* to the Amiga.

* IBM PC-AT is a registered trademark of IBM Corporation.

The aircraft flight dynamic model is a conventional six degree-of-freedom model and currently offers a choice between a large jet transport, a light aircraft, a small jet and a light, twin propeller transport. Any fixed wing aircraft can be modelled simply by adding the appropriate coefficients to the database. A first order integration scheme is implemented and an iteration rate of approximately 30 Hz is achieved. Engine modelling is simple with thrust computed as a function of the throttle setting with a transient time lag for turbojet/turbofan engines. An atmospheric turbulence model is included and severity can be controlled by the operator. This level of modelling results in a reasonably realistic simulation and serves to provide a background workload for the 'pilot'.

The operator of the controller station is provided with a mouse-driven graphical interface for selecting flight conditions, simulation configuration and for running the simulation. From the main screen the operator can access the simulator configuration screen, the flight condition screen or simply start the simulation using the default conditions and configuration. A flow diagram summarizing controller station functions is shown in figure 2.

The simulator configuration screen gives access to six main configuration settings (see configuration diagram on figure 2). The first selection determines whether the simulation will use control inputs from the sidestick, throttle and rudder pedals or pre-recorded inputs from a file. If sidestick/throttle/rudder pedal inputs are selected then the option is available to record these permitting full play-back of the simulation. The second configuration setting is the time-step for the first order integration which is generally set to reflect the iteration rate of the model. A run-time graphics option is available which displays a set of standard aircraft instruments. The operator may choose to display this screen although the iteration rate currently available is only 10 Hz. Generally this option is not used and a simple screen is displayed with the word "running" flashing on and off. Next, the operator can select whether Ethernet communications are used and the rate at which data packets are transmitted. Flight data recording is the next option available to the operator. It is possible to record flight data in a file at a rate defined by the operator. This is intended to provide data for analysis when evaluations or experiments are performed using the PC-2. Finally an audio option is available which provides engine sounds in the cockpit. These sounds have been digitized from a recording of aircraft engine sounds. The operator can choose to have the audio on or off.

The flight conditions screen is also available from the main screen (see flight conditions diagram on figure 2). From the flight conditions screen the aircraft model required for the simulation is selected. It is then possible to specify the initial conditions for velocity, heading and altitude. Wind speed and direction and turbulence severity can be defined from this screen as well.

Once the operator is satisfied with the configuration and the flight conditions then the simulation can be started from the main screen (see the start simulation diagram on figure 2). Regardless of whether the operator has chosen the instrument display or not, the level of simulation control is the same. The option to terminate the simulation is always available. In addition, turbulence severity can be varied, the simulation paused or a preset selected to re-initialise the simulation. Five presets are available which can be used to restart the simulation at any given aircraft position, altitude, speed and heading. These presets are defined in a data file called PRESET.DAT. Finally the iteration rate of the model is continuously computed and displayed on the operator's screen.

There is also the option to automatically start the simulation with the use of a startup configuration file. The controller station software looks for a file called 'FSCconfig.dat' in its current directory. If it exists the information contained in this file is used to define the flight conditions and the simulation configuration. The controller station then

proceeds to run the simulation.

In essence the controller station provides the simulation for the PC-2 and the data to animate the displays in the cockpit (i.e. via Ethernet communications). The operator has a high degree of control over the simulation and the configuration in which it is run. A stand-alone capability has been retained for developmental purposes.

4.0 RAPID PROTOTYPING ENVIRONMENT

The PC-2's instrument panel displays can be generated in several ways. Two techniques have been used to date. The first method is to use a dedicated computer program to produce the desired graphic image and then to integrate this with the PC-2 simulation using the appropriate communication protocol. The second method is to use the PC-2's rapid prototyping software package, called Virtual Application Prototyping System (VAPS)*, from Virtual Prototypes Inc. (references 5 and 6). The VAPS software runs on the Silicon Graphics IRIS Crimson VGX workstation.

Rapid prototyping allows new instruments to be created and modified easily and in much less time than the traditional hardcoding techniques. The VAPS graphical interface allows the instrument/display design team to include human factors experts and also alows end users of the prototypes (i.e. operational experts) to have some input during all stages of the design. This results in fewer design-evaluation iteration loops to reach an ergonomic and operationally effective design.

The following four stages are used in creating a VAPS prototype:

4.1 Object Editor

The first stage is the *Object Editor (OE)*. Here the user builds the prototypes graphically as well as functionally. The primitives provided by the OE are used to draw the prototypes - i.e. the static and dynamic parts of the prototypes are drawn. Once the appearance is correct, the primitives are grouped and defined as basic functional objects (e.g. a dial, barchart, button, knob, output field, etc.) in order to describe the intended operation of the instrument and the inputs it requires (reference 7). When defining the graphics primitives as functional objects the user is given a list of attributes pertaining to the particular object type being defined. For example, the object attribute list, or *property sheet*, for a dial allows the user to define the object's moving part (which could be a needle or the outer race), the center of rotation, pointer position, initial conditions, motion direction, and rotation values and limits.

There are two basic types of VAPS objects:

Output Objects take information from the simulation and display it as defined by the prototype. This type of object does not interact directly with the prototype user. The list of output objects includes: Attitude Direction Indicator (ADI), barchart, Cathode Ray Tube (CRT), cursor, dial, Guest Drawing Interpreter - for integrating independent graphics processes (GDI), light, output field, plot, Plan Position Indicator (PPI), scale, and tape.

Input Objects take user input and pass it to the VAPS sytem for processing. They are driven by the prototype user, and the data they generate may be used to drive output objects and/or communicate with the prototype logic program and external simulations. The list of input objects includes button, event label, graphical object, input field, key, knob, locator, menu, potentiometer, rotor, select, and switch.

* Virtual Prototypes Inc., 5252 de Maisonneuve West, Suite 318, Montreal, Quebec, Canada, H4A 3S5.

Objects can be embedded within other objects. This allows the VAPS user to build more complicated objects. An example of this is a dial with a variable length pointer. To construct this the VAPS user would draw the pointer and describe it as a barchart. Then the user would draw the dial face. The dial face graphics primitives and the barchart pointer would then be grouped and described as a dial with the moving part defined as the barchart pointer object. At runtime a value could be sent to the pointer object to set the length of the pointer (barchart) and to the dial object to set the dial pointer's angular position.

4.2 Integration Editor

When objects are defined in the OE each object type has associated with it data input or output *plugs*. The *Integration Editor (IE)* enables the VAPS user to connect the prototype object plugs to a data source.

An object that displays information, i.e. an output object, requires data via its plugs. An object can have 0 to 5 plugs depending on its data requirements. For example, a VAPS dial has one plug - to specify the pointer value, whereas a VAPS ADI has two plugs - pitch and roll values.

The object's plugs receive data via *channels*. These are basically data buffers which can be read from or written to by a VAPS process (see section 5.0 for more detail) and they can be connected to several sources:

Internal Stimulation allows the prototype developer to test that the objects work as intended before connecting up to the "real" data source. The objects can be connected to various data sources such as trigonometric functions (sin, cos) and simple linear functions.

Input Objects can be connected to allow direct input to output objects. For example, a potentiometer input object (driven by the prototype user at runtime) can directly drive a dial (an output object) by connecting the output plug of the potentiometer to the input plug of the dial (via a channel).

External Simulations are the most common source of data for the VAPS prototypes. The external simulation provides the data which is fed into the appropriate VAPS output objects for display or the prototypes can receive data from VAPS input objects. The user specifies what data is expected in the VAPS channel (send and receive) and what object to connect these data to (i.e. via object plugs).

4.3 Logic Editor

The prototype objects have been graphically designed and assigned a behaviour (via the OE), and connected to data (via the IE). The *Logic Editor (LE)* allows the user to specify how the prototype is to react to various events. These events can be generated either by the prototype user, via input objects (e.g. buttons, knobs etc.), or by the VAPS environment (e.g. receiving a message from another system). The LE provides the user with a tool to manage all these events - the Augmented Transition Network (ATN).

The prototype is modelled in the ATN as a Finite State Machine (FSM), with the prototype always in one of the programmed states. Each state can be made to match particular types of events, for example a button press, and respond as required by the prototype designer. VAPS provides a library of action routines that can be called up and executed in the logic program, based upon the prototype state, the prevailing conditions and the received events. The user can also incorporate code into the logic program by

inserting C-code associated with particular event matches and/or writing action routines.

The library of action routines provided by VAPS is extensive. It contains procedures for manipulating objects (e.g. hiding, appearing, flashing, moving, rotating, removing, scaling, ...); changing object plug connections; driving objects (e.g. drive dial, write output field, ...); changing attributes (e.g. set colour, set linestyle, set texture, ...); generating events; etc. (See reference 7 - VAPS Programmers Guide, Appendix A: Action Routines by Category.)

4.4 Runtime Environment

The *Runtime Environment (RE)* is the subsystem that displays and animates the objects that make up the prototype. The RE coordinates communication between all processes, frames, ATN programs and external software programs. It also synchronizes messages, and updates the screen and programs automatically.

5.0 INTER-COMPUTER COMMUNICATIONS

All computers involved in the simulation communicate via a single Ethernet backbone. In order to allow the VAPS software to be fully integrated into the system, it was decided that all communications would be implemented using existing VAPS protocols.

Data is distributed throughout the simulation using the VAPS 'channel' mechanism. A channel consists of a user-defined collection of data elements that is visible to any machine on the network having a need to access its contents. An object created using VAPS can be 'connected' to any data element from such a channel, and as a result will be automatically animated as this value changes (i.e. there need not be any hand-coding required for communications). At any time, only one process can act as a producer of the channel data, however the number of processes accessing (or consuming) the channel is conceptually unlimited. For example, the Aircraft Model acts as the producer for the 'Aircraft Data' channel, which it distributes to all of the display elements of the simulation.

At present, TCP/IP protocols are used over Ethernet, meaning each member of the simulation has a direct point-to-point socket connection with each other member. Both control and channel data messages are transmitted over the same socket connections. When the producer of channel data first begins transmission, it sends a 'BEGIN_WRITE <channel_name>' message to all other machines. If any process has an interest in this channel, it responds with the corresponding 'BEGIN_READ' message, and will receive all channel updates from then on. At any time, a process may send a 'BEGIN_READ/END_READ' message as its requirement for the channel's data changes. In addition, 'BEGIN_WRITE/END_WRITE' messages can be used at any time for the purposes of passing control of a channel to another process. Any practicable number of channels can be defined, and each is controllable separately in terms of who acts as producer/consumers and update rate. This simple protocol provides a high level of flexibility for the simulation.

The use of TCP/IP sockets has the disadvantage that any given member of the simulation must have prior knowledge of all other members on startup, so that it may connect to each in turn. In a later release of VAPS, now in hand, a simpler broadcast mechanism is used, allowing machines to connect 'on the fly' and greatly simplifing the connection process.

Once a channel has been defined as required for use with the VAPS system, Modula-2 code is generated by the MakeChannel utility. This utility has been developed at ARL to

produce Modula-2 definition and implementation modules that enable the channel to be accessed from any Amiga. Elements of the channel are implemented as members of a Modula-2 record in a definition module bearing the channel's name. At present, this module has been written so that this record always retains the most recently received data values, and any superseded data are lost. The protocol includes the possibility of having 'queued' channels where all data are kept (useful for passing key-strokes between displays, for example) and this could easily be implemented if the need arises.

6.0 DISPLAY REHOSTING

In order to maintain the rapid prototyping capability, instrumentation intended to be displayed on the Amiga screens is constructed using the VAPS tools, and rehosted onto the Amiga systems with minimal manual coding required.

The display is first constructed using the appropriate VAPS editor components, and saved as a 'frame'. A graphical metafile is then produced using the VAPS Display Specification and Rehosting Toolkit (DSRT) (reference 8) which describes the display using VPI's own text based language. This file describes the graphical representation of the display, how it should be animated, and what the requirements of each object are with respect to accessing data from channels. The VAPS Metafile language (see reference 8), is relatively high level, and producing executable code from it is not a trivial task. The metafile consists of a heirarchical list of VAPS object definitions and contains all the information needed to reproduce them. However, little assistance is given to producing compilable code that will lead to the appropriate run-time behavior of the objects, as this is highly dependent on the target system. Tools are provided in VAPS to aid in converting the metafile to code that will utilise user-written procedures to animate the display appropriately.

This conversion process has been implemented, to a limited extent, for the Amigas. Each metafile leads to the generation of a Modula-2 stand-alone module which includes functions to load and unload copies or instances of this frame (i.e. it is possible to have many independently driven copies of the same object visible on the screen at the same time: e.g. a symbol on a radar warning receiver), to turn on or off particular instances, and to cause a complete refresh of the moving parts of the display using updated driving values.

The Amiga implementation is limited by the lack of colours (a maximum of 16 in the resolutions required, as opposed to the VAPS allowance of 128 on the IRIS) and as such all colour information for each frame is presented as a series of constant definitions at the beginning of the frame's definition module. This allows 'tailoring' of the display, as complete control over colour map allocation and bit-map usage is available. In addition, the use of bit-masks for drawing operations means that non-moving parts of the display usually only have to be 'drawn' once. As a result, acceptable update rates of around 25 Hz can be achieved for most displays. Instruments needing higher update rates, or those involving complex graphical manipulation that would prove impractical for the Amigas, would most likely be hosted on the IRIS platform.

At present, only a subset of VAPS object classes has been implemented for use on the Amiga displays - it includes graphical objects, dials, scales and tapes. Further objects will be added as the requirement for them arises; it is unlikely that a full implementation will ever be constructed as some VAPS objects appear at this stage to have little relevance to likely future tasks.

7.0 PROGRAMMABLE COCKPIT APPLICATIONS

The PC-2 has many applications, from the development of novel display techniques, to prototyping and evaluating modifications to existing display instruments. It can also be used as an "idea filter" for larger, more expensive simulators. Display ideas can be prototyped in the PC-2 environment, in a relatively short time and at a low cost. Ideas that show some potential and require further investigation could then be ported onto a higher fidelity research simulator within the AOSC. Alternatively it could act as a subsiduary pilot input station in a multi-pilot simulation at the AOSC.

Some applications of the PC-2 thus far include:

7.1 Cockpit Library - PC9/A, F-111C

One of the first prototypes developed using the PC-2 rapid prototyping capability (see section 4.0) was a partial emulation of the RAAF PC9/A cockpit. The emulation included the Electronic Attitude Direction Indicator (EADI), the Electronic Horizontal Situation Indicator (EHSI), the EADI and EHSI control panels, the Engine Secondary Display Unit (ESDU), the radio panel, and some conventional instruments (VSI and Altimeter). The EADI is in fact quite different from the current PC9/A display and was designed for in-house investigations: see figure 3. The pilot's primary flight instruments for the F-111C were also prototyped for another application.

Any part of these emulations can be used separately and/or with other instruments developed. This provides a library of instruments which can be called upon to build other cockpit emulations. The PC-2 can be used to emulate a particular aircraft's cockpit or to represent a generic cockpit type.

7.2 Highway-in-the-sky

Some perspective highway-in-the-sky or Flight Director display formats, for research into effectiveness of displays for assisting in trajectory control (as in precision curved landing approaches), were developed on the IRIS 4D/85GT workstation as a stand-alone application. This was connected into the PC-2 aircraft model using the PC-2 communication protocols. Other cockpit instruments were selected from the cockpit library to enable the Flight Director to be incorporated into a cockpit for preliminary evaluation of novel features of this type of display. The centre screen in figure 4 shows the highway-in-the-sky and cockpit instruments.

7.3 F-111C Electronic Warfare Display (ALR-2002)

The rapid prototyping capability of the PC-2 was used to design and develop some new Multi-function Display (MFD) display formats for the F-111C Radar Warning Receiver (RWR). The integration of all Electronic Warfare (EW) information in the cockpit onto a single MFD was also one of the design aims of this project. Four display formats were prototyped: the current F-111C RWR format, the F/A-18 RWR format, and two new format designs. The new display formats were developed in conjunction with human factors experts of Air Operations Division and F-111C aircrew (see reference 9).

The MFD EW display prototype was then incorporated into the PC-2 cockpit (with F-111C primary flight instruments) to enable evaluation of the display formats. To provide the necessary stimulation for the EW display, an EW scenario generator (EWSG) was also produced incorporating the PC-2 moving map display. The EWSG allows EW ground threats to be located on the map with user-defined detection and lethality ranges. The PC-2 pilot then flies though the EW environment which sends EW events to the

MFD EW display. In the formal evaluation of the display, various aircrew reaction data will be recorded and analysed to facilitate comparisons between the existing and proposed new formats.

The current F-111C RWR display was also simulated on one of the Amiga stations of the PC-2. This was developed using the rapid prototyping software (VAPS) and rehosted onto the Amiga. This EWSG sends the appropriate EW event information to both of the MFD EW prototypes and the Amiga-based RWR display. This allows the old display format to be compared with the new display formats concurrently. See figure 4.

7.4 Synthetic overlays for imaging sensor displays

It is intended that the PC-2 facilities will be used for initial investigation of effectiveness of synthetic symbology formats for overlaying the displays from Forward Looking Infra-Red (FLIR), Low Light Television (LLTV) or other imaging sensors in improving aircrew awareness of terrain contours, terrain features, and obstacles, etc.

8.0 CONCLUSIONS

The PC-2 has been developed as a low-cost cockpit simulation facility and forms a part of the Air Operations Simulation Centre of Air Operations Division. It provides an environment for research into flight information integration and human interface design in the modern cockpit. The relative ease with which it can be reconfigured enables considerable flexibility in its operation and use. The PC-2 may be used, in its own right, as a tool for evaluation of new cockpit display concepts. In section 7.0 it was suggested that it could act as a prototyping tool for higher fidelity simulators within the AOSC or simply act as the platform of one player in a multi-pilot simulation.

ACKNOWLEDGEMENTS

The authors wish to recognize the contributions of others who have contributed to the development of displays for and/or elements of the PC-2. David Blunt was responsible for the development of the EW Scenario Generator and graphical interface for the PC-2 Controller Station. Grant McCabe developed the EW ground threat Scenario Generator from the moving map display. Brian Neil designed and developed the highway-in-the-sky display used in the PC-2.

REFERENCES

1. Iob, M. *Programmable Cockpit - Flight Dynamic Model,* Technical Memorandum ARL-SYS-TM-138, Aeronautical Research Laboratory, Melbourne, Australia, July 1990.

2. Craven, D.A. *Programmable Cockpit - Inter-Computer Communications and Data Flow,* Technical Memorandum ARL-SYS-TM-139, Aeronautical Research Laboratory, Melbourne, Australia, March 1990.

3. Page, A.G. *Programmable Cockpit - Head-Up Display and Outside View,* Technical Memorandum ARL-SYS-TM-137, Aeronautical Research Laboratory, Melbourne, Australia, June 1990.

4. Craven, D.A. *A Demonstration of an Aircraft Navigator's Moving Map Display,* Technical Memorandum ARL-SYS-TM-141, Aeronautical Research Laboratory, Melbourne, Australia, February 1991.

5. VAPS Conceptual Overview, Virtual Prototypes Inc., Montreal, Quebec, Canada, April 1993

6. VAPS Reference Guide, Virtual Prototypes Inc., Montreal, Quebec, Canada, April 1993

7. VAPS Programmers Guide, Virtual Prototypes Inc., Montreal, Quebec, Canada, 24 May 1991

8. DSRT Reference Guide, Virtual Prototypes Inc., Montreal, Quebec, Canada, 12 August 1991.

9. Regan, M. A., Page, A. G. and Manton, J. G. *Design of an electronic warfare display for tactical aircraft,* (ARL Report, in preparation), DSTO, Melbourne, Australia, 1994.

APPENDIX

PC-2 Hardware

The PC-2 consists of the following hardware items :

- 1 x Commodore[1] Amiga 2500 computer with PC Bridgeboard
- up to 4 Commodore Amiga 3000 computers
- 1 x Silicon Graphics[2] 4D/85 GT IRIS Workstation
- 1 x Silicon Graphics Crimson VGX IRIS Workstation
- 4 x 13 inch Commodore 1950 Monitors (640x512) all re-boxed (two with a MicroTouch[3] Capacitive Touch Screen)
- 2 x 19 inch Mitsubishi[4] High Resolution (1280x1024) Monitors both re-boxed (with one touch screen available)
- MetraByte[5] DASH-8 eight channel 12-bit Analog-to-Digital (A/D) converter
- throttle
- sidestick
- rudder pedals
- pilot seat
- framework to support monitors and controls
- two stereo speakers for audio generation.

The Amiga 2500 computer uses a Motorola[6] 68020 central processing unit and the Bridgeboard uses an Intel[7] 80286 microprocessor. This machine is used as the controller station and it hosts the flight dynamic model. The Bridgeboard controls the A/D converter which is interfaced to the throttle, sidestick and rudder pedals. Control inputs are passed to the 68020 processor for use in the flight dynamic model. Engine sounds, if required, are also generated by the controller station.

1. Commodore and Amiga are trademarks of Commodore Business Machines Inc., West Chester, Pennsylvania, USA.
2. Silicon Graphics Inc., Mountain View California, USA.
3. MicroTouch Systems Inc., Wilmington, Massachusetts, USA.
4. Mitsubishi Electric Corporation, Tokyo, Japan.
5. MetraByte Corporation, Taunton, Massachusetts, USA.
6. Motorola is a trademark of Motorola Inc., Schaumburg, Illinois, USA.
7. Intel is a trademark of Intel Corporation, Santa Clara, California, USA.

Programmable Cockpit - Stage 2

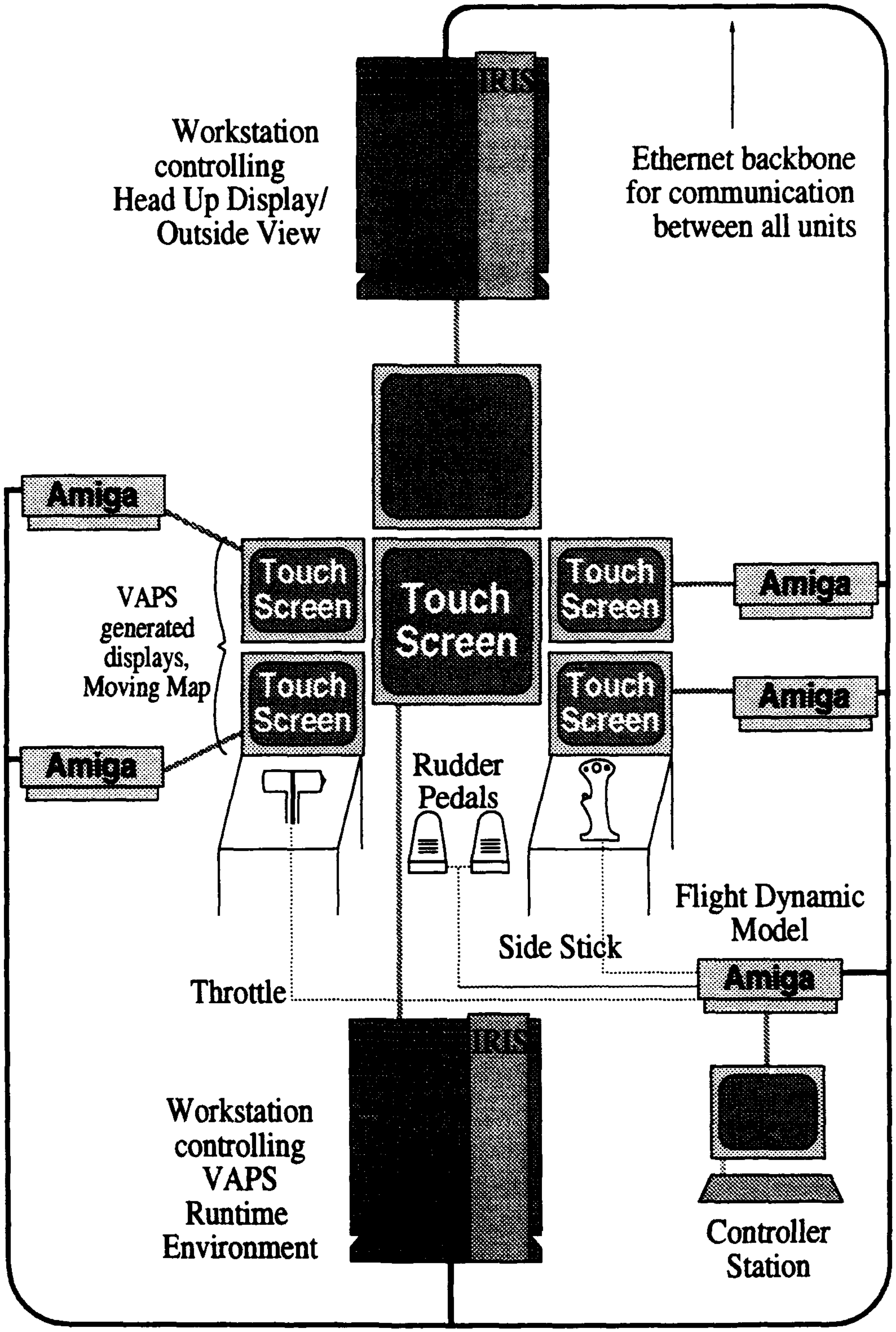

Figure 1. Programmable Cockpit - Stage 2 (PC-2)

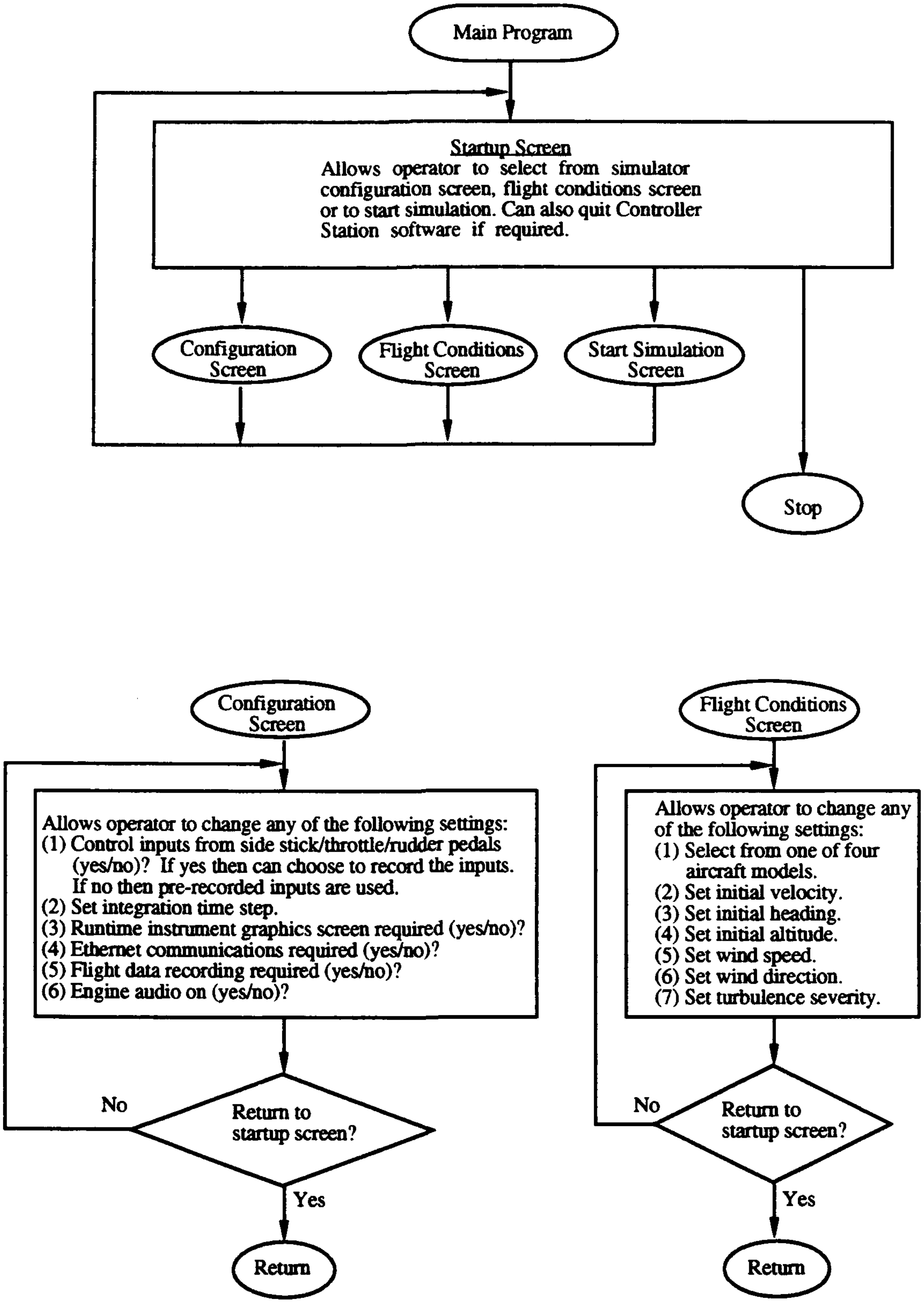

Figure 2. Flow diagram for Controller Station software

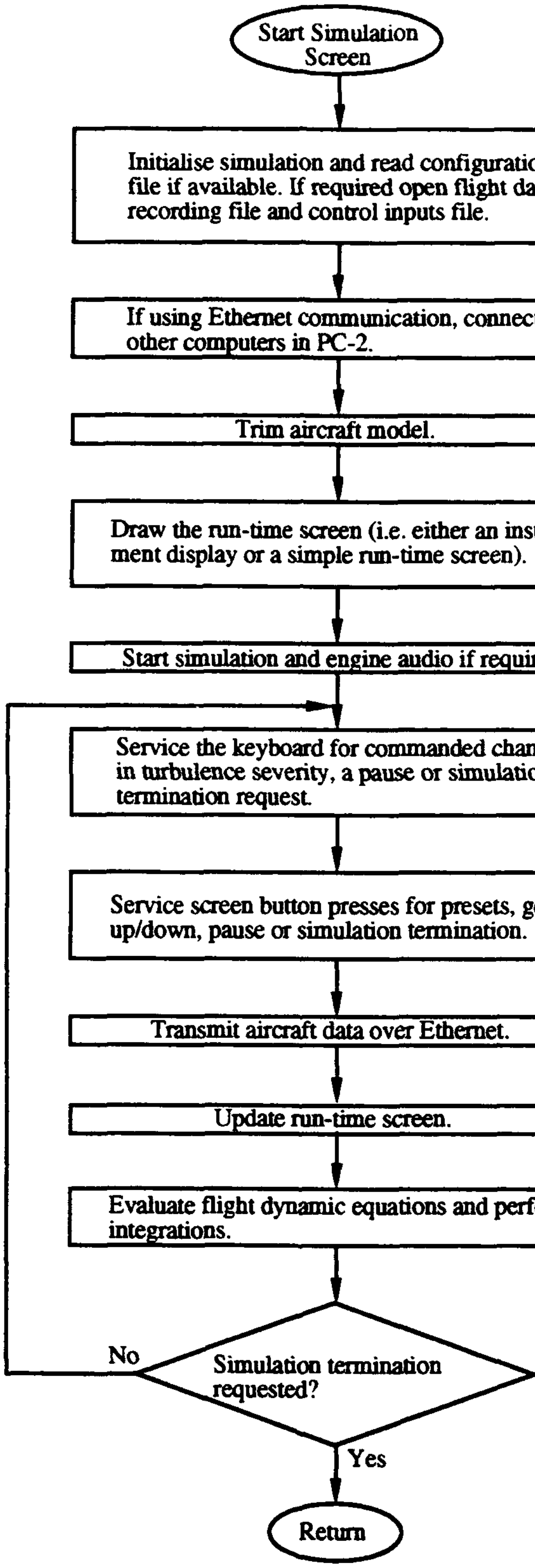

Figure 2 cont.

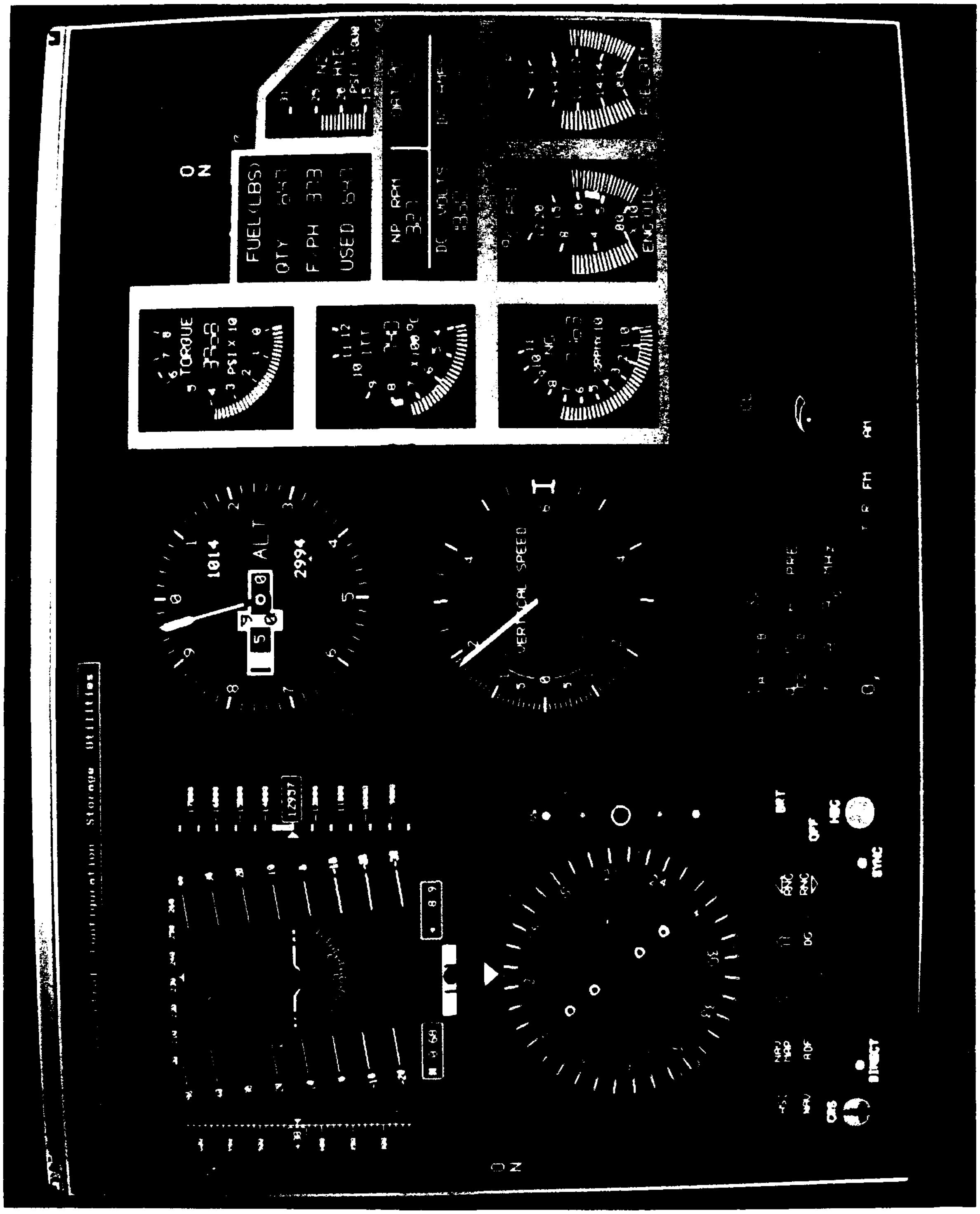

Figure 3. PC9/A cockpit emulated in VAPS

Figure 4. A typical PC-2 configuration showing highway-in-the-sky (centre screen), existing RWR display (top left screen), proposed new display (right screen) and moving map display (bottom left screen)

DISTRIBUTION

AUSTRALIA

DEFENCE ORGANISATION

Defence Science and Technology Organisation

Chief Defence Scientist
FAS Science Policy
AS Science Corporate Management
} shared copy
Counsellor Defence Science, London (Doc Data Sheet only)
Counsellor Defence Science, Washington (Doc Data Sheet only)
Senior Defence Scientific Adviser (Doc Data Sheet only)
Scientific Advisor Policy and Command (Doc Data Sheet only)
Navy Scientific Adviser (3 copies Doc Data Sheet only)
Scientific Adviser - Army (Doc Data Sheet only)
Air Force Scientific Adviser (Doc Data Sheet only)
Scientific Adviser to Thailand MRD (Doc Data sheet only)
Scientific Adviser to the DRC (Kuala Lumpur) (Doc Data sheet only)
Director Trials

Aeronautical Research Laboratory
Director
Library
Chief Air Operations Division
HAR
HAT
HHF
RL-FS
Authors: M. Iob (2 copies)
A.G. Page (3 copies)
D.A. Craven (2 copies)
P. Clark
M. Regan

Materials Research Laboratory
Director/Library

Main Library - DSTO Salisbury

Defence Central

OIC TRS, Defence Central Library
Document Exchange Centre, DSTIC (8 copies)
Defence Intelligence Organisation
Library, Defence Signals Directorate (Doc Data Sheet Only)
FAS Industry Policy and Operations

HQ ADF

Director General Force Development (Air)
DDOFFOPS
Director Analytical Studies Group

Navy

Aircraft Maintenance and Flight Trials Unit
Director Aircraft Engineering - Navy
Director of Aviation Projects - Navy
Director of Naval Architecture
Director of Occupational Health and Safety and Naval Medicine
RAN Tactical School, Library
Superintendent, Naval Aircraft Logistics
OC HMAS Albatross
OC HMAS Cerberus

Army

Engineering Development Establishment Library
Director Aviation - Army
Director of Electronic and General Procurement - Army
School of Army Aviation, Oakey
Senior Medical Officer, Oakey
CO 5 RGT Townsville
Director of Psychology - Army
US Army Research, Development and Standardization Group (3 copies)

Air Force

Aircraft Research and Development Unit
    Scientific Flight Group
    Library
RAAF Base Amberley
RAAF Base East Sale
RAAF Base Glenbrook
RAAF Base Pearce
RAAF Base Richmond
RAAF Base Tindal
RAAF Base Williamtown
DENGPP-AF
Director of Air Force Medicine
Director of Flying Safety - Air Force
Director of Psychology - Air Force
School of Technical Training
AHQ CSPT
CLSA-LC
PDR AF
OIC ATF, ATS, RAAFSTT, WAGGA (2 copies)

UNIVERSITIES AND COLLEGES

Australian Defence Force Academy
    Library
    Head of Aerospace and Mechanical Engineering

Flinders
    Library

LaTrobe
Library

Melbourne
Engineering Library

Monash
Hargrave Library

Newcastle
Library
Professor R. Telfer, Institute of Aviation

New England
Library

NSW
Physical Sciences Library

Queensland
Library, University of Queensland
Queensland University of Technology

RMIT
Library
Mr M.L. Scott, Aerospace Engineering

Sydney
Engineering Library

Tasmania
Engineering Library

Western Australia
Library

University College of the Northern Territory
Library

OTHER GOVERNMENT DEPARTMENTS AND AGENCIES

AGPS
Department of Transport & Communication, Library
Federal Airports Corporation
Civil Aviation Authority
Bureau of Air Safety Investigations

OTHER ORGANISATIONS

NASA (Canberra)
ASTA Engineering, Document Control Office
Ansett Airlines of Australia, Library
The Australian Artificial Intelligence Institute
Qantas Airways Limited
Hawker de Havilland Aust Pty Ltd, Victoria, Library
Hawker de Havilland Aust Pty Ltd, Bankstown, Library

CANADA

CANL TTCP UTP-7
CANL TTCP HTP-6

NEW ZEALAND

Defence Scientific Establishment, Library
NZNL TTCP UTP-7

UNITED KINGDOM

UKNL TTCP UTP-7
Defence Research Agency (Bedford)
G.D. Padfield

UNITED STATES OF AMERICA

USNL TTCP UTP-7
NASA Scientific and Technical Information Facility
Naval Air Warfare Center
R. Dunn
NASA Ames
N. Bucher

SPARES (56 COPIES)

TOTAL (168 COPIES)

AL 149 DEPARTMENT OF DEFENCE

DOCUMENT CONTROL DATA

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

1a. AR NUMBER	1b. ESTABLISHMENT NUMBER	2. DOCUMENT DATE	3. TASK NUMBER
AR-008-375	ARL-TN-49	MARCH 1994	A0/MGT

4. TITLE

PROGRAMMABLE COCKPIT
RESEARCH SIMULATOR

5. SECURITY CLASSIFICATION (PLACE APPROPRIATE CLASSIFICATION IN BOX(S) IE. SECRET (S), CONF (C) RESTRICTED (R), LIMITED (L), UNCLASSIFIED (U)).

DOCUMENT	TITLE	ABSTRACT
U	U	U

6. NO. PAGES
19

7. NO. REFS
9

8. AUTHOR(S)
M. IOB
A.G. PAGE
D.A. CRAVEN

9. DOWNGRADING/DELIMITING INSTRUCTIONS

Not applicable

10. CORPORATE AUTHOR AND ADDRESS

AERONAUTICAL RESEARCH LABORATORY

AIR OPERATIONS DIVISION

506 LORIMER STREET

FISHERMENS BEND VIC 3207

11. OFFICE/POSITION RESPONSIBLE FOR:

SPONSOR DSTO

SECURITY -

DOWNGRADING -

APPROVAL CAOD

12. SECONDARY DISTRIBUTION (OF THIS DOCUMENT)

Approved for public release.

OVERSEAS ENQUIRIES OUTSIDE STATED LIMITATIONS SHOULD BE REFERRED THROUGH DSTIC, ADMINISTRATIVE SERVICES BRANCH, DEPARTMENT OF DEFENCE, ANZAC PARK WEST OFFICES, ACT 2601

13a. THIS DOCUMENT MAY BE ANNOUNCED IN CATALOGUES AND AWARENESS SERVICES AVAILABLE TO ....

No limitations

14. DESCRIPTORS
Cockpit simulators
Cockpit displays
Computer graphics
Workstations
Dynamic models
User interface

15. DISCAT SUBJECT CATEGORIES
0104

16. ABSTRACT

The 'Programmable Cockpit' is a low-cost facility to be used for study and development of the aircrew-vehicle interface for future aircraft systems. It was designed so that instrument layouts and display formats could be reconfigured rapidly and tested in a realistic aircraft representation or emulation, with the pilot under representative workload conditions. It is not intended to be used for pilot training and/or evaluation.

It uses personal computers and computer graphics workstations linked together to represent the aircraft displays, and includes representative flight dynamic models for fixed-wing aircraft. Controls include a sidestick, rudder pedals, throttle, and touch sensitive screens.

PAGE CLASSIFICATION
UNCLASSIFIED

PRIVACY MARKING

THIS PAGE IS TO BE USED TO RECORD INFORMATION WHICH IS REQUIRED BY THE ESTABLISHMENT FOR ITS OWN USE BUT WHICH WILL NOT BE ADDED TO THE DISTIS DATA UNLESS SPECIFICALLY REQUESTED.

16. ABSTRACT (CONT).

This document gives a brief overview of the complete Programmable Cockpit, including the controller station (which incorporates the flight dynamic model), inter-computer communications, rapid prototyping environment and display rehosting.

17. IMPRINT

AERONAUTICAL RESEARCH LABORATORY, MELBOURNE

18. DOCUMENT SERIES AND NUMBER	19. WA NUMBER	20. TYPE OF REPORT AND PERIOD COVERED
Technical Note 49	76000B	

21. COMPUTER PROGRAMS USED

22. ESTABLISHMENT FILE REF.(S)

23. ADDITIONAL INFORMATION (AS REQUIRED)